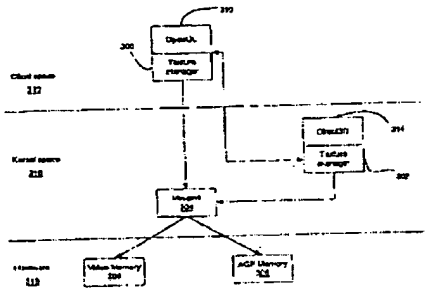
REMARKS

The Examiner has again rejected Claims 1-29 under 35 U.S.C. 102(e) as being anticipated by Nguyen et al. (US2002/0101427A1), hereinafter "Nguyen." Applicant respectfully disagrees with this assertion.

Specifically, the Examiner now relies on the following excerpts from Nguyen to make a prior art showing of applicant's claimed "sending an instruction request to memory utilizing a texture module in a graphics pipeline" and "receiving instructions from the memory in response to the instruction request utilizing the texture module in the graphics pipeline" (emphasis added). See this or similar language in each of the independent claims.

> [0025] The processes of the present invention are performed by processor 202 using computer implemented instructions, which may be located in a memory such as, for example, main memory 204, memory 224, or in one or more peripheral devices 226-230.
>
> [0031] Turning next to FIG. 4, a flowchart of a process used for managing texture memory is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in FIG. 4 may be implemented in a texture manager, such as texture manager 300 or 302 in FIG. 3. This process is applied to both video memory and memory in the data processing system for storing textures.
>
> [0032] The process begins by receiving a request to store a texture in texture memory (step 400). A texture is reloaded as needed (step 402). Texture memory is then allocated to the current texture in the request (step 404). In this example, the allocation occurs through a call made by the texture manager to a miniport or other memory allocation mechanism. A determination is made as to whether the allocation was successful (step 406). In this example, allocations are done a per mipmap basis. A texture object consists of one or more mipmaps starting from level 0 and the allocation always begins with mipmap 0, then 1, and so on. A mipmap is a reduced resolution version of a texture map, used to texture a geometric primitive whose screen resolution differs from the resolution of the source texture map.
>
> [0036] The removed texture is marked (step 412). If a texture is removed from APG memory, this texture is marked as a APG texture such that when it is reactivated or loaded again, this texture will be loaded into APG memory. If the texture is removed from video memory, it is marked as a video texture so that this texture will be reloaded into video memory the next time it is requested for use. Memory is allocated to the current texture object (step 414) with the process then returning to step 406 as described above.

Figure 3



Specifically, with respect to applicant's claimed "sending an instruction request to memory utilizing a texture module in a graphics pipeline" (emphasis added), the Examiner argues "using computer implemented instructions located in memory corresponding to definition of IEEE dictionary for instruction which is binary word sending serially into device, texture will be loaded into AGP memory corresponding to sending instructions to memory."

First, it is somewhat unclear as to the Examiner's intention regarding the mention of "corresponding to the definition of IEEE dictionary for instruction which is binary word sending serially into device." There is absolutely no mention of the IEEE dictionary in Nguyen. If the Examiner is attempting to make some sort of Official Notice argument or the like, applicant formally requests a specific prior art showing of ALL of applicant's claimed features in any future action. Note excerpt from MPEP below.

> "If the applicant traverses such an [Official Notice] assertion the examiner should cite a reference in support of his or her position." See MPEP 2144.03.

Further, it appears that the Examiner is attempting to establish that Nguyen suggests "computer implemented instructions located in memory," and "texture will be loaded into AGP memory." Whether or not Nguyen discloses such, it is clear that the Examiner is relying on Nguyen's "texture manager" to meet applicant's claimed

"texture module." However, Nguyen's texture manager, itself, does **not** send an instruction request to memory. Just because Nguyen's processor uses computer implemented instructions located in memory, such teaching does not rise to the level of specificity where a texture module sends an instruction request to memory.

Still yet, with respect to applicant's claimed "sending an instruction request to memory utilizing a texture module in a graphics pipeline" and "receiving instructions from the memory in response to the instruction request utilizing the texture module in the graphics pipeline" (emphasis added), the Examiner argues that Nguyen suggests "using computer implemented instructions located in memory, the process begins by receiving a request to store a texture in texture memory." The mere disclosure of "receiving a request to store a texture in texture memory" simply does not rise to the level of specificity present in applicant's claims.

Assuming that the Examiner is attempting to equate the aforementioned "request" with applicant's claimed received "instructions," applicant responds by pointing out that, whether the Examiner's assertion is true or not, this suggestion fails to meet applicant's claimed receipt of the instructions in response to the instruction request utilizing the texture module. This is evidenced by Nguyen's disclosure that the receipt of the "request" is the beginning of the related process (thus implying there is no step prior). To this end, under the Examiner's interpretation of the prior art, Nguyen simply does not meet applicant's claimed preceding operation, namely "sending an instruction request to memory utilizing a texture module in a graphics pipeline."

It thus appears that the Examiner is erroneously equating the higher-level functionality of the processor and related hardware of Nguyen and the underlying modules, etc. thereof, and inappropriately using such disparate entities interchangeably in a failed attempt to meet applicant's claims.

Despite the foregoing paramount differences and in the interest of expediting the prosecution of the present application, applicant has amended each of the independent claims to emphasize that the instructions are received from video memory (emphasis added).

Again, it emphasized that only applicant teaches and claims retrieving instructions from video memory, as claimed, which includes any memory including graphics data that is accessible to graphics hardware.

Just as important, it appears that the Examiner has simply dismissed applicant's previous arguments regarding the deficiencies in the Examiner's rejection of the remaining claims. Such arguments are reiterated below.

Just by way of example, applicant claims in dependent claims that "the instructions are adapted for controlling a texture environment module coupled to the texture module" (see Claim 6). The Examiner relies on the following excerpt from Nguyen to show such feature in the prior art.

> [0025] The processes of the present invention are performed by processor 202 using computer implemented instructions, which may be located in a memory such as, for example, main memory 204, memory 224, or in one or more peripheral devices 226-230.

After a careful review of such excerpt, applicant contends that this excerpt is lacking, especially in view of the shortcomings of the Examiner's application of Nguyen to the independent claims. For example, the abovementioned "processor" does not include a "texture module," as claimed by applicant. Only applicant teaches and claims sending "instructions requests" and retrieving "instructions" "utilizing a texture module" which provides the aforementioned advantages that are non-existent in the prior art including Nguyen.

Similarly, the Examiner relies on the following excerpt from Nguyen to make a prior art showing of applicant's claimed "initial instructions [that] control at least the sending of the instruction request by the texture module" (see Claim 9), "a complete instruction set [that] is received in response to the instruction request" (see Claim 18), and "a partial instruction set [that] is received in response to the instruction request" (see Claim 19).

> [0032] The process begins by receiving a request to store a texture in texture memory (step 400). A texture is reloaded as needed (step 402). Texture memory is then allocated to the

> current texture in the request (step 404). In this example, the allocation occurs through a call made by the texture manager to a miniport or other memory allocation mechanism. A determination is made as to whether the allocation was successful (step 406). In this example, allocations are done a per mipmap basis. A texture object consists of one or more mipmaps starting from level 0 and the allocation always begins with mipmap 0, then 1, and so on. A mipmap is a reduced resolution version of a texture map, used to texture a geometric primitive whose screen resolution differs from the resolution of the source texture map.

Once again, there is simply no mention in the excerpt above of any sort of instruction retrieval utilizing a texture module, let alone "initial instructions [that] control at least the sending of the instruction request by the texture module," "a complete instruction set [that] is received in response to the instruction request," and "a partial instruction set [that] is received in response to the instruction request," as claimed by applicant.

Still yet, applicant brings to the Examiner's attention the following additional independent claim that has been added, which includes at least a portion of the allowable subject matter mentioned hereinabove:

"30. (New) A method for retrieving instructions from video memory utilizing a cache in a graphics pipeline, comprising:

sending an instruction request to video memory in a graphics pipeline; and

receiving instructions from the video memory in response to the instruction request utilizing a cache in the graphics pipeline."

Again, the aforementioned anticipation criteria has simply not been met by the Examiner's reference. A specific prior art showing of such claim limitations or a notice of allowance is respectfully requested. All of the pending independent claims are thus deemed allowable along with any claims depending therefrom.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. For payment of any fees due in connection with the filing of this paper, the

Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NVIDP064/P000286).

Respectfully submitted,

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172-1120
Telephone: (408) 505-5100